

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

AD-A270 825



tion is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and reviewing the collection of information, Send comments regarding this burden estimate or any other aspect of this reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson 2, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

2. REPORT DATE

3. REPORT TYPE AND DATES COVERED

FINAL/01 AUG 90 TO 31 MAR 93

4. TITLE AND SUBTITLE

VLSI IMPLEMENTATION OF NEUROMORPHIC
LEARNING NETWORKS (U)

5. FUNDING NUMBERS

7013/DARPA
F49620-90-C-0042

6. AUTHOR(S)

Dr Joshua Alspector

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Bellcor
290 West Mt Pleasant Ave
PO Box 486
Livingston NJ 07039-04868. PERFORMING ORGANIZATION
REPORT NUMBER

AFOSR-TR- 93 0754

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR/NM
110 DUNCAN AVE, SUITE B115
BOLLING AFB DC 20332-000110. SPONSORING/MONITORING
AGENCY REPORT NUMBER

F49620-92-C-0042

DTIC
ELECTE
OCT 18 1993
D
S
E

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

12b. DISTRIBUTION CODE

UL

13. ABSTRACT (Maximum 200 words)

The researchers have shown how to rigorously derive deterministic systems from stochastic ones in the Boltzmann machine framework that they are using for their implementations. They have further shown how to search for new learning algorithms suitable for VLSI implementation using a genetic algorithm approach. They have analyzed the effect of precision constraints such as is found in hardware implementations on the learning and generalization abilities of neural networks. They have studied the learning behavior of neural networks under conditions where they where they can or cannot classify perfectly.

14. SUBJECT TERMS

15. NUMBER OF PAGES

8

16. PRICE CODE

17. SECURITY CLASSIFICATION
OF REPORT
UNCLASSIFIED18. SECURITY CLASSIFICATION
OF THIS PAGE
UNCLASSIFIED19. SECURITY CLASSIFICATION
OF ABSTRACT
UNCLASSIFIED20. LIMITATION OF ABSTRACT
SAR(SAME AS REPORT)

COVER SHEET

VLSI Implementation of Neuromorphic Learning Networks
Contract Number F49620-90-C-0042, DEF
P.I. - Joshua Alspector, Bellcore

Final Report

ARPA Order: 7013
Program Code: 8D10,9D10
Name of Contractor: Bellcore
Effective Date of Contract: 1Jun90
Contract Expiration Date: 31May92
Amount of Contract Dollars: 370,389
Contract Number: F49620-90-C-0042
Principal Investigator: Joshua Alspector
and Phone Number: (201) 829-4342
Program Manager: Steve Suddarth
and Phone Number: (202) 767-4931
Short Title of Work: VLSI Implementation of Neuromorphic Learning Networks

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. 7013
Monitored by AFOSR Under Contract No. F49620-90-C-0042

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government

422341

93-24267



93 1 14 076

SUMMARY

VLSI Implementation of Neuromorphic Learning Networks
Contract Number F49620-90-C-0042, DEF
P.I. - Joshua Alspector, Bellcore

Final Report

1. Technical Problem

We wish to extend our study of neural-style learning in electronic systems to a usefully large scale. Our long term goal is to define and develop an electronic learning system suitable for solving real-world problems using learning by example.

2. Methodology

The study of electronic implementation issues will be extended to large scale systems using a three pronged approach: A) Further development of learning algorithms and architectures suitable for modular VLSI implementation. B) Functional simulation of large scale systems using benchmark test problems. C) Design and fabrication of prototype chips suitable for inclusion in and testing of such systems.

3. Technical Results

3.1 Theory

We have shown how to rigorously derive deterministic systems from stochastic ones in the Boltzmann machine framework that we are using for our implementations. We have further shown how to search for new learning algorithms suitable for VLSI implementation using a genetic algorithm approach. We have analyzed the effect of precision constraints such as we find in hardware implementations on the learning and generalization abilities of neural networks. We have studied the learning behavior of neural networks under conditions where they can or cannot classify perfectly. We have defined and shown how to use a measure to determine when either a stochastic or deterministic system has settled and will use this measure in our electronic system. For further details, see Section 1 of the final report.

3.2 Simulation

Our results show that the Boltzmann machine learning we use in our VLSI implementation gives approximately the same performance as the more popular back-propagation algorithm used in most simulations. Furthermore, both algorithms scale up to large size similarly. We have started a study of perturbative learning for implementable feed-forward neural networks. For further details, see Section 2 of the final report.

3.3 Implementation

We have designed, fabricated, and tested an experimental prototype of a large learning microchip containing 32 neurons and 496 bidirectional synapses. The chip settles either stochastically (using a novel electronic noise generator) or deterministically (using variable gain neuron amplifiers). Learning experiments on a single chip microsystem show results similar to what we obtained in simulation. The chip is capable of running at 100,000 patterns per second (100 million connections per second per chip) and of being cascaded to form systems of larger size. We have completed the design and fabrication of a synapse-only chip to enhance cascadability. For further details, see Section 3 of the final report.

4. Further Research

We would eventually like to interface our simulator to a multi-chip learning system and add other functions such as mean-field content-addressable memory.

We would like to fully realize the potential of our learning microchips by incorporating the chip set into a large multi-chip, VME based learning microsystem.

We would like to find a suitable means of learning in analog VLSI for feed-forward and dynamic neural networks.

We would like to apply these techniques to challenging, real-time problems such as image classification.

5. Special Comment on Integrated Circuit Technology for Neural Networks

Significant progress in the implementation of large, multi-chip, electronic learning systems is hampered by the state of current VLSI technology. An integrated circuit technology which can create small (about 1 square micron) learning synapses is highly desirable. Some modification of the current analog floating gate technology might be suitable. Furthermore, multi-chip systems would be most easily achieved if a suitable wafer-scale integration technology were available. Neural networks are a natural candidate for using such a technology.

FINAL REPORT

**VLSI Implementation of Neuromorphic Learning Networks
Contract Number F49620-90-C-0042, DEF
P.I. - Joshua Alspector, Bellcore**

Final Report

1. Neural Network Theory

1.1 Accomplishments

Several authors have recently proposed deterministic learning algorithms as approximations to learning in stochastic systems. We have studied^[1] two deterministic learning algorithms, and showed how they may be viewed as different ways of performing the approximation to the fully stochastic system, which in this case was the Boltzmann machine. We focused in particular on the representation of probability distributions in the deterministic systems and related them to the true distributions. Specifically, if one takes the Boltzmann machine probability distribution and uses a saddle point approximation, one gets the usual mean-field equations. However, if one uses the mean-field approximation that the correlations factorize, one obtains the algorithm of Pineda.

From the point of view of hardware applications it would be very useful to devise learning algorithms which require only a limited precision, such as using only binary synaptic weights and neural states. One way of doing this is to take existing learning algorithms and discretize the weights during learning. We have taken an alternative approach of using genetic algorithms to search the space of all possible algorithms.^[2] In the case of a single layer perceptron with binary weights, we have shown that we get a well known algorithm devised by other means, namely, the directed drift algorithm of Venkatesh. We are now trying to extend these ideas to the multi-layered case, about which much less is known.

We have studied learning and generalization in single-layer feedforward networks, whose weights are constrained to take on a discrete set of values.^[3] As far as we know this is the first analytic study of the effect of weight precision (important for hardware implementations) on the learning and generalization ability of neural networks. Our analytic results are obtained within the replica approach, which is verified through Monte Carlo simulations. It is shown that, depending on the architecture of the network and on the source of the training examples, three qualitatively different behaviors emerge. This distinction, which is manifested through the dependence of the training and generalization errors on the size of the training set, suggests a possible way to determine the suitability of the architecture to the learning task. We conjecture that this distinction is relevant to the more interesting case of multi-layered networks.

We have calculated the training and generalization errors of three well known learning algorithms using methods of statistical physics.^[4] We focus in particular on inconsistent algorithms which are unable to perfectly classify the training examples, and show that the asymptotic behavior of these algorithms is different from the case of consistent algorithms. Our results are in agreement with bounds derived by computational learning theorists. We also demonstrate that one of the algorithms studied performs almost indistinguishably from the Bayes learning algorithm, while having the advantage of being implementable in a single-layer network. This last point is important if one is to systematically evaluate the performance of learning systems, and compare them to standard statistical approaches.

In feedback neural networks, especially for static pattern learning, a reliable method of settling is required. Simulated annealing has been used but it is often difficult to determine how to set the annealing schedule. Often the specific heat is used as a measure of when to slow down the annealing process, but this is difficult to measure. We have proposed another measure, volatility,^[5] which is easy to measure and related to the Edwards-Anderson model in spin-glass physics. We have been studying the usefulness of this measure in simulations of dynamics in Boltzmann and mean-field networks, and have shown how to use it to speed up learning. We have established a theoretical basis for the volatility measure to substitute for the specific heat in annealing. Simulations have verified the validity of this measure and shown how to use it to speed up annealing and learning. This quantity is far easier to measure than specific heat because only the knowledge of neural states and not the weights are needed. This seems to hold promise for being an easy to measure way of controlling the noise and gain in our neural network chips.

2. Neural Network Simulation

2.1 Accomplishments

We presented a paper^[6] at the Neural Information Processing Systems (NIPS) conference in November, 1990. The paper shows, by simulation of benchmark test problems such as NETtalk, that network learning algorithms of the type we are implementing (Boltzmann and mean-field) work as well as the far more commonly used back-propagation technique. Since some form of feedback connections are required so that the teacher signal on the output neurons can modify weights during supervised learning, we argue that full time feedback, as opposed to the part-time feedback of back-propagation, is more plausible to investigate, even for static pattern learning where the dynamics of recurrent connections are not utilized fully. Relaxation methods are needed for learning static patterns with full-time feedback connections. Feedback network learning techniques have not achieved wide popularity because of the still greater computational efficiency of back-propagation. We show by simulation that relaxation networks of the kind we are implementing in VLSI are capable of learning large problems just like back-propagation networks. The availability of hardware learning should give a boost to these methods. Our benchmark problems are parity, replication, and NETtalk.

We presented a paper^[7] at the Neural Information Processing Systems (NIPS) conference in December, 1992. This described a parallel, stochastic method for learning in feed-forward networks without doing back-propagation of errors. The work focused on a perturbation technique that *measures*, not *calculates*, the gradient. Since the technique uses the actual network as a measuring device, errors in modeling neuron activation and synaptic weights do not

cause errors in gradient descent. Simulations showed that the method learns and scales well. We used the benchmark problems of parity, replication, contiguity, and hamming coding to check scaling properties. It appears that we can exploit the parallel nature in an implementation to achieve a speedup over computer simulation. This is a component of a current proposal to ARPA for VLSI implementation of learning for use in image classifiers.

3. Neural Network Implementation

3.1 Accomplishments

We have designed, fabricated, and performed functional and learning tests on an experimental prototype of a 32 neuron learning microchip.^[8] This 160,000 transistor chip also contains 496 bi-directional synapses and a 32 channel uncorrelated noise generator. We have measured the transfer functions of the analog neuron and the analog portions of the synapse and demonstrated variable gain. We have verified the functionality of the digital portion of the synapse. We have also shown that the noise generator works and demonstrated its effect on the neuron transfer function. We have built a set of test boards for performing learning using an experimental prototype single chip. We have integrated the chip, boards, data generators and analyzers, and an X-windows based interface into a learning system based on the chip.

More importantly, we have demonstrated that the chip learns. This work follows up on the previous year's simulation paper by performing Boltzmann and mean-field experiments in actual learning hardware rather than just simulations. The results obtained are similar to those obtained in software. Measurements show that the potential learning speed of the hardware is 100,000 patterns per second roughly independent of the problem size. If all the synapses on the chip were utilized, this would imply a learning speed of 100 million connection updates per second (CUPS) per chip. This is roughly 10,000 times faster than Sparc 2 simulations of Boltzmann learning.

We have completed design and fabrication of an experimental prototype of a 1024 bi-directional synapse chip to enhance cascability of the above neuron-containing chip.^{[9] [10]} This chip has been tested for functionality and appears to work well. The two chips together can form the building blocks for a much larger neural system.

We presented a paper^[11] at the 1992 IEEE Neural Networks for Signal Processing Workshop at Elsinore, Denmark, September, 1992. This showed how our Boltzmann and mean-field prototype chip can be used for content addressable memory with a capacity far larger than ordinary Hopfield memory by using mean-field settling and hidden units. Results show good agreement between simulations and the actual chip. A possible use of this technique would be for vector quantization or other coding.

3.2 Other Deliverables

Video available (delivered to B. Yoon, Seattle, July 11, 1991); Joshua Alspector and Anthony Jayakumar, "Bellcore Neural Learning System - The Video"; A learning neural microchip which settles using both electronic noise and variable gain neurons is described. The functions of the various components are demonstrated on oscilloscopes. An integrated software-hardware system which can learn by example is demonstrated. An example learning task performed by the chip is displayed in an X-windows based software system. Videotaped on location in Room 2E-377, Bellcore, Morristown, NJ, June 24-25, 1991. Copyright Bellcore 1991.

Microchip available (delivered to B. Yoon, Seattle, July 11, 1991)

Assorted view-graphs including both the neuron-containing and synapse-only learning chips are available.

REFERENCES

1. R. Meir, "On Deriving Deterministic Learning Rules From Stochastic Systems", *International Journal of Neural Systems* 2, (4), 283-289 (1992).
2. J.F. Fontanari and R. Meir, "Evolving a Learning Algorithm for the Binary Perceptron", *NETWORK* 2 (4), 353-360 (1992).
3. R. Meir and J.F. Fontanari "Learning from Examples in Weight Constrained Neural Networks", *J. Phys. A* 25, 1149-1168 (1992).
4. R. Meir and J.F. Fontanari "Calculation of learning curves for inconsistent algorithms", *Phys. Rev. A* 45 (12), 8874-8884 (1992).
5. J. Alspector, S. Luna, and T. Zeppenfeld, "A Volatility Measure for Annealing in Feedback Neural Networks", *Neural Computation*, Vol.4, No. 2, (March 1992) pp. 191-195
6. J. Alspector, R.B. Allen, A. Jayakumar, T. Zeppenfeld, & R. Meir "Relaxation Networks for Large Supervised Learning Problems" in *Advances in Neural Information Processing Systems 3*, R.P. Lippmann, J. Moody, and D. Touretzky (eds.), Morgan-Kaufmann, Palo Alto, (1991), p.1015 (Proceedings NIPS '90, Denver, CO)
7. J. Alspector, R. Meir, B. Yuhas, A. Jayakumar, D. Lippe, "A Parallel Gradient Descent Method for Learning in Analog VLSI Neural Networks," in *Advances in Neural Information Processing Systems 5*, (San Mateo, CA: Morgan Kaufman Publishers, 1993) pp. 836-844.
8. J. Alspector, A. Jayakumar, S. Luna, "Experimental Evaluation of Learning in a Neural Microsystem", presented at IEEE Conf. on Neural Information Processing Systems, Dec 2 - 5, 1991, Denver, CO and in J.E. Moody, S.J. Hanson and R.P. Lippmann, (eds.) "Advances in Neural Information Processing Systems 4" Morgan Kaufmann Publishers, San Mateo, CA (1992), pp. 871-878.
9. A. Jayakumar and J. Alspector, "A Cascadable Neural Network Chip Set with On-Chip Learning using Noise and Gain Annealing", presented at the Custom Integrated Circuits Conference, Boston, MA, May, 1992 pp. 19.5.1-19.5.4.
10. A. Jayakumar and J. Alspector, "On-Chip Learning in Analog VLSI using Simulated Annealing", in *Proceedings of the Government Microelectronics Applications Conference*, (Las Vegas, NV, November 10-12, 1992) pp. 277-280
11. J. Alspector, A. Jayakumar, B. Ngo, "An Electronic Parallel Neural CAM for Decoding", in S.Y. Kung, F. Fallside, J.Aa. Sorenson, C.A. Kamm, (eds.) *Neural Networks for Signal Processing II*, Proceedings of the 1992 IEEE Workshop, (Elsinore, Denmark, September, 1992) pp. 581-587